LOW COST SYMBOL ERROR CORRECTION CODING AND DECODING

Inventor: Chin-Long Chen, Fishkill, NY (US)
Assignee: International Business Machines Corporation, Armonk, NY (US)

Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

Appl. No.: 08/062,894
Filed: May 17, 1993

Related U.S. Application Data
Continuation of application No. 07/318,983, filed on Mar. 6, 1989, now abandoned.

Int. Cl. H03M 13/11 (2006.01)
U.S. Cl. 714/752
Field of Classification Search 371/37.1, 371/37.6, 40.1, 39.1, 714/752
See application file for complete search history.

References Cited
U.S. PATENT DOCUMENTS
3,601,798 A 8/1971 Hsiao
4,358,848 A 11/1982 Patel
FOREIGN PATENT DOCUMENTS
EP 0300139 * 1/1989

OTHER PUBLICATIONS

ABSTRACT
Byte or symbol organized linear block codes are optimized in terms of reducing the number of ones in their parity check matrices by means of symbol column transformations carried out by multiplication by non-singular matrices. Each optimized symbol column preferably, and probably necessarily, includes a submatrix which is the identity matrix which contributes to low weight check matrices and also to simplified decoding procedures and apparatus. Since circuit cost and layout area are proportional to the number of Exclusive-OR gates which is determined by the number of ones in the check matrix, it is seen that the reduction procedures carried out in accordance with the present invention solve significant problems that are particularly applicable in the utilization of byte organized semiconductor memory systems. Reduced weight coding systems are also generated in accordance with weight reducing procedures used in conjunction with modified Reed Solomon codes. The decoding and encoding methods and apparatus are extendable to the inclusion of any number of check symbols.

19 Claims, 14 Drawing Sheets
OTHER PUBLICATIONS


* cited by examiner
FIG. 6A

$S_1 = S_{1,1}$
$S_2 = S_{1,2}$
$S_3 = S_{1,3}$
$S_4 = S_{1,4}$
$S_5 = S_{1,5}$
$S_6 = S_{2,1}$
$S_7 = S_{2,2}$
$S_8 = S_{2,3}$
$S_9 = S_{2,4}$
$S_{10} = S_{2,5}$
$S_{11} = S_{3,1}$
$S_{12} = S_{3,2}$
$S_{13} = S_{3,3}$
$S_{14} = S_{3,4}$
$S_{15} = S_{3,5}$

FIG. 6B

$H_j$

$H_{1j}$

$H_{2j}$

$H_{Mj}$

$b$ BITS

$b$ BITS

$b$ BITS
FIG. 6D

RECEIVED SYMBOL BITS

CORRECTED SYMBOL BITS FOR SYMBOL J

FIG. 6E

FROM SYMBOL ERROR DETECTION UNITS

SYNDROME NON-ZERO

UNCORRECTABLE ERROR INDICATOR
FIG. 8A2

RECEIVED SYMBOL BITS

- $S_{k1} = s_6$
- $S_{k2} = s_7$
- $S_{k3} = s_8$
- $S_{k4} = s_9$
- $S_{k5} = s_{10}$

$D_1$ $D_2$ $D_3$ $D_4$ $D_5$

- $D_1$ $D_2$ $D_3$
- $D_4$ $D_5$

CORRECTED SYMBOL BITS FOR SYMBOL #1

TO UNCORRECTABLE ERROR DETECTOR 3D
FIG. 8B2

RECEIVED SYMBOL BITS

D4

D1

21.1

Sk1 = S6

AND

21.2

Sk2 = S7

AND

21.3

Sk3 = S8

AND

21.4

Sk4 = S9

AND

22.1

D2

22.2

D3

22.3

D4

22.4

CORRECTED SYMBOL BITS FOR SYMBOL #1

TO UNCORRECTABLE ERROR DETECTOR 30
LOW COST SYMBOL ERROR CORRECTION CODING AND DECODING

This is a continuation of application(s) Ser. No. 07/318,983 filed on Mar. 6, 1989, now abandoned.

BACKGROUND OF THE INVENTION

The present invention is generally directed to the encoding and decoding of binary data so as to employ error correction coding (ECC) methods for symbol error correction and detection. Apparatus for decoding such codes is also provided and implemented in a fashion which exploits code structure to provide encoders and decoders which minimize circuit cost, particularly with respect to the number of Exclusive-OR gates needed. More particularly, an embodiment employing three check symbols is presented.

The utilization of error correction and detection codes in electronic data processing and transmission systems is becoming more and more important for several reasons. In particular, increased problem complexity and security concerns require ever increasing levels of reliability in data transmission. Furthermore, the increased use of high density, very large scale integrated (VLSI) circuit chips for use in memory systems has increased the potential for the occurrence of soft errors such as those induced by alpha particle background radiation effects. Furthermore, the increased use of integrated circuit chips has led to memory organizations in which each output memory word is derived from multiple bit patterns received from a plurality of different circuit chips. Accordingly, it has become more desirable to be able to protect memory system integrity against the occurrence of chip failures since such failures produce multi-bit errors which are associated with a single chip. It is thus seen that it is desirable to employ error correction coding systems which take advantage of the memory system organization itself, particularly with respect to minimizing the probability of error. For this reason, it is desirable to employ codes which are based upon symbols or bytes of data. (It is noted that, as used herein, the term "byte" does not necessarily refer to an 8-bit block of binary data but rather is more generic.)

As noted, error correcting codes have been applied to semiconductor memory systems to increase reliability, reduce service costs and to maintain data integrity. In particular, single error correcting and double error detecting (SEC-DED) codes have been successfully applied to many computer memory systems. These codes have become an integral part of the memory design for medium and large systems throughout large portions of the computer industry.

However, the error control effectiveness of an error correction code depends on how the memory chips are organized with respect to the code. In the case of single error correcting and double error detecting codes, the one bit per chip organization is the most effective design. In this organization, each bit of a code word is stored in a different chip, thus any type of failure in a chip can corrupt at most one bit of the code word. As long as the errors do not line up in the same code word, multiple errors in the memory are correctable in this scheme.

As the trend in chip design has continued toward higher and higher density, it has become more difficult to design one bit per chip types of memory organizations because of the system granularity problem. For example, the system capacity has to be at least four megabits if one megabit chips are used to design a memory with a 32 bit data path. However, in a b bit per chip memory, a chip failure may result in from 1 to b bit errors, depending upon the type of failure. A failure could be a cell failure, a word line failure, a bit line failure, a partial chip failure or a total chip failure. With a maintenance strategy that allows correctable errors in the memory to accumulate, SEC-DED codes are not particularly suitable for b bit per chip memories. Since multiple bit errors are not correctable by SEC-DED codes, the uncorrectable error rates can be high if the distribution of chip failure types is skewed to those types that result in multiple bit errors.

A more serious problem is the loss of data integrity due to the misregistration of some multiple errors. It is therefore seen that for byte organized memory systems, it is desirable to employ single byte error correcting and double byte error detecting (SBC-DBD) codes in b bit per chip memories. With such codes, errors generated by a single chip failure are always correctable and errors generated by a double chip failure are always detectable. Thus, the uncorrectable error rate can be kept low and the data integrity can be maintained.

Discussions concerning codes that are suitable for this purpose can be found in the article "Error-Correcting Codes for Byte-Organized Memory Systems", by Chin-Long Chen in Volume IT-32, No. 2 of the IEEE Transactions on Information Theory (March 1986).

In the construction of error correcting codes contrary criteria are often present. The first criteria relates to the error correcting capabilities of the code itself. The second criteria relates to the ease of encoding and more particularly, of decoding the information received. More complex codes often require more complicated circuitry for decoding. With the constraints of electronic circuit chip "real estate" being at a premium with respect to circuit design, it is seen that it is greatly desirable to be able to implement both encoding and decoding circuits utilizing as few circuit gates as possible with each gate having a minimum number of attached signal lines. Accordingly, one of the objects of the present invention is the construction of minimum cost encoders and decoders without the corresponding sacrifice of correction and detection capabilities.

In particular, one of the classes of codes which is particularly applicable to the present invention is a code employing three check symbols. Such a code can be described as an extended Reed Solomon code having a parity check matrix of the form:

\[
H = \begin{bmatrix}
    1 & T^a & T^{2a} & \ldots & 1 & 0 & 0 \\
    1 & T^{a+1} & T^{2(a+1)} & \ldots & 0 & 1 & 0 \\
    1 & T^{a+1} & T^{2(a+1)} & \ldots & 0 & 0 & 1 
\end{bmatrix}
\]  

In Equation 1 above, \( H \) is the parity check matrix, a well known concept in error correction coding. \( I \) represents the identity matrix and \( O \) represents a matrix all of whose elements are zero. In general, \( T \) is a matrix with \( b \) rows and \( b \) columns and represents the companion matrix of a \( b^i \) degree primitive polynomial over the field GF(2). In one particular embodiment of this form of Reed Solomon code, the integer \( a \) is taken to be zero. Further information concerning such codes is found in the aforementioned article by the present inventor.

SUMMARY OF THE INVENTION

In accordance with a preferred embodiment of the present invention, a method for encoding binary data which occurs in a block of \( a \) bits and subblocks of at most \( b \) symbol bits, comprises subjecting the \( a \) bits of data to parity check conditi-
tions determined by a binary parity check matrix which is either equal to or derivable from a matrix of the form \( H = [H_1, H_2, \ldots, H_M] \), where each symbol column \( H_j \) is a column of \( M \) disjoint binary submatrices \( H_j \) having \( b \) rows and at most \( b \) columns and wherein at least one of the \( N \) symbol columns possesses a minimal number of ones. In particular, the present inventor has realized that the submatrices in each symbol column may be multiplied by non-singular binary matrices without affecting error correction or detection properties. Nonetheless at the same time the resultant code exhibits a reduction in the number of ones in the parity check matrix. It is seen that this is a desirable goal since the number of \( 2 \)-way Exclusive-OR gates required to generate the \( j \)th check bit is equal to the number of ones in the \( j \)th row of the binary matrix \( H \) less one. Thus, in the situation in which each symbol possesses \( b \) bits, the total number of Exclusive-OR gates required to generate all check bits is equal to the number of ones in the binary matrix \( H \) \( -3b \) (for \( M = 3 \)), \( M \) being the number of check symbols. In addition, the number of ones in a symbol column of \( H \) is proportional to the number of Exclusive-OR gates required to decode syndrome errors. Thus, to reduce error correction encoding and decoding hardware, it is seen it is desirable to reduce the number of ones in each symbol column, \( H_j \).

Thus, in accordance with the present invention it is possible to manipulate symbol columns of \( H \) through the utilization of multiplication by non-singular matrices which thus create equivalent codes. While multiplication of the \( H \) matrix itself is known to effect the creation of equivalent codes having identical error correction and detection capabilities, this method does not appear to have been applied to symbol columns in the parity check matrices of byte organized codes.

In the process of symbol column manipulation taught by the present invention, it is further desired to carry out non-singular transformations for symbol columns independently of one another in such a way that one of the \( M \) disjoint binary submatrices which comprise column \( H_j \) is an identity matrix. It is not known, but it is suspected, that it is a necessary condition for minimality that at least one of the \( M \) disjoint submatrices is in fact an identity matrix. It is clearly seen that this is a desirable goal in that an identity matrix itself possesses a minimal number of ones and cannot in fact possess fewer ones without a reduction in rank. Thus, in accordance with preferred embodiments of the present invention, the symbol columns in the parity check matrices all possess at least one identity submatrix. Furthermore, it is seen that this situation is always possible to create, particularly with respect to the situation involving Reed Solomon codes. The constraint of having an identity matrix present further simplifies the design of decoding circuitry. Although each symbol decoding unit must be designed separately, nonetheless, there is a decided saving in the number of Exclusive-OR gates required, sometimes resulting in a reduction in circuitry by as much as approximately 25%. Furthermore, the separate design requirement is only a factor during code design and does not affect coding or decoding cost or speed.

In accordance with yet another embodiment of the present invention, a method for decoding linear binary codes of the kind indicated above comprises the steps of determining a syndrome vector \( S \) for a received binary vector \( v \) with the syndrome \( S \) being treated as having \( M \) subvectors \( S_1, S_2, \ldots, S_M \). Then, for at least one symbol position \( j \) the \((M-1)\) matrix-vector products \( H_jS_k \) \((i=k)\) are determined. In the above, \( k \) denotes the matrix row of \( H \), which is an identity matrix. The value of \( k \) depends on \( j \) and varies from symbol column to symbol column. The \((M-1)\) vector products are then compared with the corresponding syndrome vectors \( S \) to determine the presence or absence of correctable symbol errors. Lastly, at least the \( j \)th symbol is corrected using the vector pattern \( S_j \) as an error pattern whenever an indication of a correctable symbol error arises from the comparison operation. A corresponding apparatus for decoding such linear binary codes is also presented herein.

Accordingly, it is an object of the present invention to permit the design of parity check matrices having a minimal weight, that is having a minimal number of 1's in their structure.

It is also an object of the present invention to extend the applicability of at least a subset of Reed Solomon codes.

It is yet another object of the present invention to construct error correcting codes and encoding apparatus which consume minimal amounts of circuitry, particularly as measured with respect to area consumed on VLSI circuit chips.

It is a still further object of the present invention to construct decoding apparatus in which the aforementioned matrix-vector product operations and comparison operations are carried out in a single layer of Exclusive-OR gates.

It is also an object of the present invention to construct codes, encoders and decoding systems which are particularly applicable to byte organized memory systems without in any way impairing the error correction or detection capabilities of such codes.

It is yet another object of the present invention to be able to construct codes which are derivable from minimal weight codes so as to further simplify coder and decoder design and implementation.

Lastly, but not limited hereto, it is an object of the present invention to implement low cost error correcting codes which enhance computer storage integrity, reliability and serviceability.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the concluding portion of the specification. The invention, however, both as to organization and method of practice, together with the further objects and advantages thereof, may best be understood by reference to the following description taken in connection with the accompanying drawings in which:

FIG. 1A illustrates a matrix for a byte organized code having three check symbols and which is illustrative of the kinds of parity check matrices which may be manipulated and processed in accordance with the present invention for the purpose of circuit and decoder simplification;

FIG. 1B illustrates a matrix \( T \) which is the companion matrix for a specific primitive polynomial \( p(x) \) which is used herein in an exemplary manner and which may be substituted in the matrix of FIG. 1A;

FIG. 1C illustrates a parity check matrix of a more general form exhibiting in particular a larger number of check symbols and further illustrating the extension of such parity check matrices to codes employing a higher number of check symbols;

FIG. 2 illustrates a parity check matrix which has been manipulated in accordance with the present invention;

FIG. 3 illustrates a parity check matrix constructed in accordance with one embodiment of the present invention in which one initially employs a longer length code with a greater number of symbols and selectively reduces symbol columns from the larger parity check matrix in accordance with the requirements of minimal weight.

FIG. 4 (in parts A and B) is the binary matrix representation of the parity check matrix shown in matrix form in FIG. 3;
FIG. 5 is an overall block diagram of a decoding apparatus in accordance with the present invention;
FIG. 6A illustrates the segmentation of syndrome vector S into a plurality of M subvectors, S;
FIG. 6B illustrates the organization of the symbol columns H_j of the parity check matrix H;
FIG. 6C illustrates the organization of multiplication and comparison circuitry for the particular case of symbol j and
the presence of M=3 check symbols;
FIG. 6D illustrates the circuitry employed for correcting bits in symbol j;
FIG. 6E illustrates the circuitry which provides uncorrectable error indications;
FIG. 7 (in parts A and B) is a binary check matrix which represents a column reduced form of FIG. 4;
FIG. 8A (in parts A1 and A2) illustrates the particular circuitry employed to implement the matrix vector product and comparison operations performed for symbol j=1 for the
matrix shown in FIG. 4;
FIG. 8B (in parts B1 and B2) is similar to FIG. 8A but is more particularly directed to the reduced parity check matrix
seen in FIG. 7.

DETAILED DESCRIPTION OF THE INVENTION

The description which follows first considers the situation in which the number of check symbols in the code, M,
equals 3. However, in the latter part of the present description, attention is directed to the general case in which M
is greater than 3.

In one embodiment, the present invention is directed to a
class of binary codes which have been found to be useful
because of their error correction properties and because of
their ease of decoding. However, the present invention
improves upon these codes with respect to both encoding and
decoding, particularly when measured against circuit cost as
determined by the number of Exclusive-OR gates employed.
The present invention is specifically directed to code construction based upon codes having a parity check matrix
organized in the form of a plurality of symbol columns. Each
of the symbol columns preferably includes submatrices which are powers of a given companion matrix, T, which
is the matrix associated with a primitive polynomial over the
Galois Field GF(2). For example, if the primitive polynomial
p(x) is of degree b and has the following form:

\[ p(x) = x^4 + x^3 + x + 1 \]

then the companion matrix T is of the form given below:

\[
T = \begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 0 & 0 & p_1 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & p_{b-1}
\end{bmatrix}
\]

It is noted that in the mathematical field at hand the elements,
namely 0 and 1, are their own additive inverses since addition
is carried out in a modulo 2 fashion. This fact accounts for the
absence of minus signs which would otherwise normally be
present in the form given for the T matrix for other fields.
Generally, the companion matrix T possesses b rows and b
columns. The matrix T is also always non-singular. However,
certain code simplifications and reductions, which are more
particularly described below, can alter this result, at least
formally, when specific binary parity check matrices are
derived by column or row deletion. Appropriate parity check
matrices are developed by computing the various powers of
the companion matrix T and inserting such matrices in the
positions shown, for example as in Equation (1) above. The
result is a binary parity check matrix with specific symbol
error correction properties in which each symbol is repre-
sented by at most b bits.

In one particular embodiment of the present invention, it is
convenient to consider parity check matrices in which the
number M of check symbols is 3. Since the submatrices, T,
each generally include b rows and b columns and because of
the structure of the code and the primitive nature of the
polynomial p(x), there are at most 2^b such matrices (not
counting the T^0 which is equal to the identity matrix).
Thus it is useful, at least initially, to consider parity check
matrices having the following form:

\[
H = \begin{bmatrix}
I1 & I & \ldots & I & 0 & 0 \\
I & T & \ldots & T^m & 0 & 0 \\
I & T^2 & \ldots & T^n & 0 & 0 \\
\end{bmatrix}
\]

In the above m (which is unrelated to M) is less than or equal
to 2^b–2. Also in the above I represents in general, an identity
matrix with b rows and b columns the matrix O is a bxb matrix
with all zero elements. The code defined by Equation (4)
requires 3b binary check bits whose positions correspond to
the last 3b binary columns of H (for the matrix shown). It is
further noted that the matrix above is of the form H=H_1 + H_2 + \ldots + H_m, where N is the number of code symbols, that is the
number of blocks of b bits in a code word. In this respect, as
long as each symbol is represented by b bits, the number of
bits n in a code word is Nb. In the matrix H above, each of the
N columns represents a binary b column matrix. These
columns are referred to herein as symbol columns. Also, the
weight of a binary column is the number of 1’s in the column
and the weight of a symbol column is the number of 1’s in the b
constituent binary columns of the symbol column.

One of the observations that is key to an understanding of
the present invention is the fact that each symbol column H_j
can multiply a bxb non-singular matrix which has the effect
of forming a different code by scrambling the bits in a symbol
column in an invertible linear fashion. The code characteri-
tics however are not changed. This manipulation is possible
since the code is being treated as a symbol code having N
separate b bi blocks. The individual bit blocks are indepen-
dently manipulable in this fashion. In fact, the manipulation
of the symbol columns of H by multiplication by non-singular
matrices is not dependent on the symbol columns of H having
any specific form. As indicated, it is applicable to any parity
check matrix for byte organized codes.

The motivation for this matrix multiplication arises from the
fact that the number of two-way Exclusive-OR gates required
to generate i-th check bit is equal to the number of 1’s in the
i-th row of the binary matrix of H, less 1. Thus the total
number of Exclusive-OR gates required to generate all check
bits is equal to the number of 1’s in the binary matrix of H, less
3b. It is thus seen that to reduce error correction coding
hardware, it is desirable to reduce the number of 1’s in each
symbol column and in general to reduce the number of 1’s in
the binary matrix H itself. In general, the parity check matrices
H shown in Equations (1) or (4) are not optimal in this sense.
In general, each symbol column is multiplied by all \( b \times b \) non-singular binary matrices and the resulting symbol column having the lowest number of 1’s is thereafter employed as the symbol column of choice. Thus, in the present invention symbol columns are employed which exhibit optimality in the sense that each symbol column possesses a minimum number of 1’s. Such symbol columns and the parity check matrices comprised of such symbol columns are optimal in the sense that they exhibit minimal hardware requirements.

It is suspected, though it is not certain, that general optimization in this fashion produces symbol columns which exhibit an identity submatrix somewhere in the stack of disjoint submatrices that comprise the symbol column. However, in its most general embodiment, the present invention is not specifically limited to this requirement. Nonetheless, for purposes of decoding and for establishing error patterns, it is highly desirable to be able to construct parity check matrices with symbol columns which do include an identity submatrix. As seen below, this problem is particularly easily solved in the case of parity check matrices of the kind illustrated in Equations (1) or (4) above.

In particular, attention is now more specifically directed to optimization methods by parity check matrix modification which can be carried out on matrices of the specific form shown in Equation (4). In particular, it is seen that, apart from the first column and the last three symbol columns, each symbol column includes an identity matrix in the first matrix row and matrices of the form \( T^q \) and \( T^{-q} \) in the other two matrix rows. Because of the structure of the matrix \( T \), it is in itself non-singular and maybe employed to perform the non-singular symbol column transformations indicated above. For example, it is possible to multiply symbol column \( H_2 \) in the matrix from Equation (4) by either \( T \) or \( T^{-1} \) to form symbol columns having either of the following structures respectively:

\[
H_1 = \begin{bmatrix} T & \tilde{T}^q & I \\
T^2 & 1 & I \\
I & 1 & T^{-2} \end{bmatrix}
\]

\[
H_2 = \begin{bmatrix} T^{-1} & \tilde{T}^{-q} & I \\
I & 1 & T^{-2} \\
1 & 1 & T^{-2} \end{bmatrix}
\]

In each case it is seen that the resultant symbol column \( H_2 \) includes an identity matrix. Nonetheless, the three possible symbol columns \( H_\lambda \) indicated in Equations (4), (5a) and (5b) do not necessarily possess the same weight. Nonetheless, it is possible without changing code characteristics to select any one of these three possibilities as symbol column \( \# 2 \) for parity check matrix \( H \). This is a key aspect to understanding the operation and effect of the present invention. In particular, the present invention provides a low cost symbol error correction coding scheme which has a minimal weight in at least one and preferably in each symbol column of its parity check matrix. It is further noted that it is not necessary to have an identity matrix as the first non-zero submatrix of a symbol column in the parity check matrix. As long as there is an identity matrix somewhere in the symbol column, the error pattern for the erroneous symbol can always be readily obtained. This observation is key to the construction of minimal weight symbol columns in the parity check matrix and a key to the construction of low cost decoding circuitry. It is also seen that the matrix structure in Equation (4) is particularly applicable to the construction of codes in which an identity matrix is always present in a symbol column.

The above discussion leads to a simple algorithm for the construction of improved parity check matrices. Such matrices produce lower cost encoders and are particularly amenable to the production of lower cost decoders especially in the situation in which an identity matrix is present in each symbol column. More particularly, for each symbol column \( H_\lambda \) of the form:

\[
H_\lambda = \begin{bmatrix} I \\
T^q \\
T^{-q} \end{bmatrix}
\]

where \( q \) is an integer, the following procedure may be employed to improve the check matrix \( H \) by modification of the symbol columns. In particular, let \( W_1 \) be the weight of \( H_\lambda \). Next, multiply the matrix \( T^{-q} \) by the column \( H_\lambda \), and let \( W_2 \) be the weight of this modified symbol column. Next, multiply the matrix \( T^q \) by the symbol column \( H_\lambda \) and let \( W_3 \) be its corresponding weight, that is the number of 1’s in the resultant matrix. Next, the 3 weights \( W_1, W_2 \) and \( W_3 \) are compared. If \( W_3 \) is the smallest, then \( H_\lambda \) is replaced by \( H_\lambda T^{-q} \). Similarly, if \( W_2 \) is the smallest weight, then the symbol column \( H_\lambda \) is replaced by the symbol column \( H_\lambda T^q \). It is noted that these symbol column manipulations are preferably carried out for as many symbol columns as possible. It is also noted that a symbol column substitution is not necessarily made in all instances. That is to say, \( W_1 \) will not always be greater than \( W_2 \) and \( W_3 \).

The above principles are illustrated in the following example in which \( b=5, m=14 \) and \( p(x)=1+x^2+x^5 \). In this case, the companion matrix \( T \) is given as:

\[
T = \begin{bmatrix} 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \end{bmatrix}
\]

This matrix is also illustrated in FIG. 1B. The parity check matrix of Equation (4) is expressed in terms of the powers of \( T \) as shown in FIG. 1A. The matrix of all zeroes is expressed as \( O \). The weight distribution for the 18 symbol columns of the parity check matrix \( H \) shown in FIG. 1A is computed to be (15 17 19 22 26 31 32 35 37 36 38 37 35 34 5 5 5). The total weight of the parity check matrix is therefore 466. Applying the weight comparison algorithm given above, an improved parity check matrix results and is shown in FIG. 2. The weight distribution for the symbol columns in this new matrix is (15 17 19 22 26 31 32 33 35 32 30 35 34 31 28 24 5 5 5) and the total weight of this new matrix is therefore seen to be 424 which represents a 9% reduction in weight. It is also particularly important to note that in the parity check matrix shown in FIG. 2, each symbol column includes at least one identity matrix. It is also seen that for the first 7 symbol columns no
symbol column substitutions are made. This is seen to be in agreement with the comments presented above concerning the size of $W_i$ relative to $W_2$ and $W_3$.

Before considering the possibilities of achieving even further reductions, it is useful to consider the matrix shown in FIG. 1C. In this particular matrix, the number of check symbols, $M$, is 4. This matrix and the matrix shown in Equation (1) also indicate the general mechanism for extending the results of the present invention to codes having a greater number of check symbols.

With further regard to FIG. 1A, it also should be noted that the symbol columns are indeed of the form shown in Equation (6). In particular, because of the periodic nature of the powers of T, $T^{-1}$, $T^{-3}$, $T^{-5}$, $T^{-28}$, etc. this fact is relevant in comparing symbol column $H_Q$ with Equation (6). Similar statements are applicable to the other symbol columns.

With particular attention focused on matrix row #2 from FIG. 1A, it is seen that it is possible to extend the number of symbol columns in H by adding powers of $T^2$. Again, because of the periodic nature of the powers of $T^2$, it is possible to have such matrices range all the way up to $T^{17}$ where $m=2^7-2$. If $b=5$, $m=30$. Suppose that $H_{max}$ is a parity check matrix having the maximal number of symbol columns. This affords the opportunity for even greater reductions. In particular, in practical applications, the code length is usually less than the maximum possible value for a given code class. In particular, the number of symbols required for the code may be less than N. In this case, it is possible to select a subset of symbol columns from $H_{max}$ to form the desired parity check matrix. In order to minimize the number of Exclusive-OR gates required in a hardware coding or decoding implementation, these symbol columns are chosen which have a minimal weight. This leads to yet a new method for parity check matrix construction. In particular, the first step of this method involves the application of the matrix multiplication and weight selection algorithm described above but now with $m$ having its maximum value $2^6-2$. Then from a modified $H_{max}$ the desired number of symbol columns are selected so as to form a subset having the smallest total weight to use as a reduced parity check matrix.

These principles can be applied to the example presented above in which $b=5$, $N=18$ and $p(x)=1+x+x^2+x^3$. This is the same set of parameters used in FIGS. 1A and 1B. Applying the second reduction method of the present invention (selection from the columns of $H_{max}$), an even further reduced parity check matrix is found. This matrix is shown in FIG. 3 wherein the matrix $T$ is the same as described above. The binary form of this matrix is shown in FIG. 4 (parts A and B). The binary form of the matrix is seen below to be useful in designing decoder circuitry for receiving code words which have been subjected to the parity check conditions specified by the matrix $H$ in FIG. 4. The total number of $1$'s in this matrix is 350. This now represents a 25% reduction as compared to the matrix shown in FIG. 1 which has a weight of 466.

Although the considerations provided above have generally only been directed to codes having three check symbols, the methods of the present invention can be applied to constructing a byte organized error correcting code with minimum weight symbol columns in its parity check matrix for any number of check symbols. In particular, it is seen in FIG. 1C that by multiplication by an appropriate power of $T$ it is always possible to form a symbol column in which there is present an identity matrix. It is suspected that this is sufficient to guarantee an absolute minimum of $1$'s in a symbol column. In short, it is conjectured that it is only necessary to employ as non-singular multiplicand matrices, the various distinct powers of $T$. In general, there are $2^n-2$ such powers of $T$ to consider. Nonetheless, in the more general case, it is possible to explore the results obtained by using all possible non-singular $n$ by $n$ matrices. While this problem becomes much more difficult with higher numbers of symbol bits $b$, the problem is nonetheless tractable when $b=3$, 4 or 5. Nonetheless, it is seen that since each symbol column can be manipulated independently of the others, the multiplication and simplification operations are nonetheless straightforward and do not grow rapidly with the number of symbol columns or with the number of check symbols.

Attention is now specifically directed to the advantages of the present invention which are applicable to decoding. Certainly, it is seen that for encoders, the smaller the number of $1$'s in the parity check matrix the smaller will be the amount of hardware required to implement it since the number of two-way Exclusive-OR gates required with the $i^{th}$ check bit is equal to the number of $1$'s in the $i^{th}$ row of the binary matrix, minus one. With respect to decoders, the situation is in general not as simple. However, because of simplifications brought about by the presence of an identity matrix as one of the submatrices in each symbol column, it is possible to exploit certain code structures to provide decoders having a minimal number of Exclusive-OR gates. Furthermore, such decoders are readily designable using automated circuit layout algorithms and methods.

In particular, suppose that $E$ is the $b$ bit error pattern for a symbol. The positions of errors are indicated by the presence of $1$'s in the $E$ vector. Suppose that the $j^{th}$ symbol of a code word contains $E$ as the error pattern for that symbol. The error syndrome of $E$ is equal to the product of the $j^{th}$ column of the parity check matrix $H$ and the vector $E$. The syndrome $S$ is given as $Hy$, where $y$ is the received code word vector. Thus, if the code vector $x$ is sent and the vector $y$ is received then $x-y=E$ or $y-x+E$. However, it is known that $Hy=HE$ because $x$ is a code word for which $Hx=0$. In particular, in this situation in which there are three check symbols, the syndrome $S$ can be partitioned into three subvectors of $b$ bits each. This partitioning is particularly illustrated in FIG. 6A. Thus, $S=(S_1, S_2, S_3)$. Suppose further that the three submatrix components of the $j^{th}$ symbol column of $H$ are denoted by $H_{jj}$, $H_{j2}$, and $H_{j3}$. This organization is particularly illustrated in FIG. 6B for the general case where $M$ is not necessarily equal to 3. In this case, when all the errors occur in a single symbol position, the matrix equation $S=HE$ can be restated as three separate matrix equations as follows:

$$S_1=H_{j1}E \quad (8a)$$
$$S_2=H_{j2}E \quad (8b)$$
$$S_3=H_{j3}E \quad (8c)$$

In the above, $S_1$, $S_2$, and $S_3$ and $E$ are considered as column vectors and $E$ is the $b$ bit error pattern for symbol $j$ from the $n$ bit long $E$ vector. For each and every symbol column $j$ one of the submatrices $H_{jj}$, $j=1, 2, 3$, represents an identity matrix and accordingly one of either $S_1$, $S_2$, or $S_3$ is the error pattern $E$ for symbol $j$. Which one of the $S_i$ is the identity matrix depends upon $j$, the particular symbol column being considered. However, this is known from the structure of the original parity matrix $H$ which is used to encode the vector which has been received. Thus, the error pattern $E$ can be readily iden-
tified as one of the three subvector components of \( S \). Once \( E \) has been identified, the symbol error status is determined by checking that all three congruence relations of Equation (8) are true.

For example, consider the code defined by the parity check matrix in FIGS. 3 and 4. If the errors are in the first symbol position, then the error pattern \( E \) is equal to \( S_2 \) since \( H_{11} \) is an identity matrix in the first symbol column. To verify if the errors are really in the first symbol position, it is necessary to check that \( S_1^T = T^H S_2 \) and that \( S_2^T = T^H S_2 \). While this computation must be carried out for each symbol column separately, it is nonetheless seen that there are relatively easy ways of carrying out the computation and the necessary comparison.

Attention is now specifically directed to decoding methods and apparatus for code words constructed in accordance with certain specific embodiments of the present invention. In particular, FIG. 5 illustrates an overall block diagram for error correction and detection for a symbol organized error correction code. FIGS. 6C, 6D and 6E illustrate more detailed aspects of the system shown in FIG. 5.

With respect to FIG. 5, it is noted that decoding is done separately for each symbol. Nonetheless, in general, all syndrome bits are available to symbol error detection units 1 through 4. Each such symbol error detection unit 10 essentially performs a binary matrix multiplication operation and a comparison operation. The output of functional block 10 is essentially an enable signal supplied to a corresponding one of the bit error detection and correction units 20. Correction units 20 receive an enable signal from a corresponding symbol error detection unit 10 and each unit 20 also receives a specific subset of syndrome data bits. In each case, the subset of syndrome bits received matches the syndrome subvector having the same relative position in the syndrome as the identity matrix has in the associated symbol column from \( H \). Additionally, for each separate symbol, bit error detection and correction units 20 receive the symbol data bits themselves. Functional block 20 therefore typically receives b syndrome bits and b symbol data bits. Additionally, functional block 20 receives the enable signal from symbol error detection unit 10. This enable signal is also supplied to uncorrectable error detection unit 30. Each of these units is more particularly described below in the discussions relating to FIGS. 6C through 6E.

Attention is now specifically directed to symbol error detection unit 10. Such a unit is shown in more detail in FIG. 6C for the situation in which \( M = 3 \). In particular, only one such unit is shown namely that unit which is associated with the \( j \)-th symbol. In the case shown, the syndrome is partitioned into three groups of \( b \) bits each. Two matrix vector product operations are performed, one of which computes the product \( H_1 S_{2} \), where \( i \) is not equal to \( k \) and where \( k \) depends on \( j \). The other matrix product computed by unit 10 is the matrix vector product \( H_1 S_k \) where \( 1 \) is not equal to \( k \), which depends on \( j \). It is noted that each symbol error detector 10 is associated with its own unique value of \( k \). In particular, \( k \) is the identifier of the matrix row in the symbol column which includes the identity matrix. For the first symbol column in the parity check matrix shown in FIG. 4, \( k = 2 \). For the second symbol column for this matrix, \( k = 3 \). For the first symbol column \( j = 1 \) and the two matrix products that are actually computed are \( H_1 S_2 \) and \( H_3 S_2 \). More particularly, if we also denote the syndrome vector \( S \) as \( (s_1, s_2, \ldots, s_{15}) \) then symbol column multiplier 12 is actually seen to carry out the following two matrix operations given by Equations (9a) and (9b) below:

\[
H_{11} S_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_2 \\ s_3 + s_5 \\ s_7 + s_8 + s_{10} \end{bmatrix}
\]

\[
H_{11} S_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_2 \\ s_3 + s_4 \\ s_7 + s_8 + s_{10} \end{bmatrix}
\]

These last two equations, particularly when expressed in terms of Exclusive-OR or modulo 2 operations, clearly indicate that circuit complexity is directly proportional to the number of \( 1 \)'s in the parity check matrix. These equations also directly specify circuitry which implements the matrix products indicated. Bear in mind that Equations (9a) and (9b) represent the situation only for the matrix of FIG. 4 and for the case in which \( j = 1 \) (symbol \( \# 1 \)) for which it is seen from FIG. 4 that \( k = 2 \). The two resultant vectors from Equations (9a) and (9b) are compared with subvectors \( S_1 \) and \( S_2 \) respectively. This operation is carried out by means of comparators 13 and 14 respectively as seen in FIG. 6C. However, since the field operation of addition is equivalent to an Exclusive-OR function and since a Boolean Exclusive-OR function also performs an implicit comparison upon its input variables, it is actually seen that the comparison in multiplication may in fact be carried out by means of a single layer of multi-input Exclusive-OR gates. This situation is more clearly seen in FIG. 8A which is discussed below. With respect to symbol error detection unit 10 however, the comparison results are supplied to NOR-gate 15 which provides the desired enable signal to a bit error detection and correction unit and to uncorrectable error indicator 30.

Bit error detection and correction units are more specifically illustrated in FIG. 6D. There, error pattern bits from syndrome subvector \( S_k \) namely bits \( (S_2, S_3 \ldots, S_{15}) \) are provided to AND gates 21. AND gates 21 receive the enable signal from symbol error detection unit 10 so as to permit correction of bits for that symbol position only. Thus, the error pattern represented by syndrome subvector \( S_k \) is supplied under appropriate conditions directly to Exclusive-OR gates 22. These Exclusive-OR gates receive symbol data bits which are corrected by applying the error pattern (syndrome subvector) to Exclusive-OR gates 22 which operate as conventional inverters. Thus, where necessary, received data bits are inverted in accordance with the error pattern. This is made possible by the presence of an identity matrix in each of the symbol columns of \( H \). Thus, bit error detection and correction unit 20 operates to produce corrected symbol bits for symbol \( j \), as shown.

Attention is next directed to uncorrectable error indicator 30 which receives enable signals from symbol error detection units 10 which are thus supplied to multiple input NOR-gate 31. The output of this NOR-gate is directed to AND gate 32 which passes this signal whenever a non-zero syndrome is detected. See FIG. 6E.

The relation between decoding hardware and the submatrices of a parity check matrix \( H \) is particularly illustrated in FIG. 8A. FIG. 8A illustrates a symbol error detection unit 10 and bit error detection and correction unit 20 for the first symbol position \( (j = 1) \) associated with the parity check matrix.
of FIG. 4. In particular, functional block 10 of FIG. 8A implements the matrix multiplication shown above in Equations (9a) and (9b). In particular, attention is first directed to functional block 10 and Exclusive-OR gates 16.1 through 16.5 which carry out the matrix multiplication operations shown in Equation (9a). The matrix-vector multiplication operation (actually a sequence of modulo 2 summations) is carried out using the syndrome bits supplied to Exclusive-OR gates 16.1 through 16.5 which are shown having both short lead lines. The short lead lines in FIGS. 8A and 8B are thus associated with the matrix multiplication operation. Additionally, the layer of Exclusive-OR gates shown is capable of simultaneously carrying out a comparison operation because of the inherent nature of the Exclusive-OR function (namely its being the inverse of the equivalence function; NOR gate 17 complementing this inverse). Thus, in order to compare the matrix vector product \( H_s S_z \) with the syndrome vector \( S_z \), it is seen that it is only necessary to supply to the syndrome subvector bit positions \( s_1 \) through \( s_7 \) to Exclusive-OR gates 16.1 through 16.5 respectively. (Be sure to distinguish \( S_z \), a b bit syndrome subvector from \( S_z \), the first syndrome bit.) This is shown using the longer input lead lines. Thus the multiplication and comparison operations indicated in FIG. 6C are actually performed in a single Exclusive-OR gate layer as shown. At the same time, the short line input signals supplied through Exclusive-OR gates 16.6 through 16.10 effectively carry out the matrix multiplication indicated in Equation (9b) namely the matrix-vector product \( H_y S_z \). This result is effectively compared with syndrome subvector \( S_z \) by supplying syndrome bit positions \( s_{11} \) through \( s_{15} \) to Exclusive-OR gates 16.6 through 16.10 respectively. The Exclusive-OR gate outputs (gates 16.1 through 16.10) are supplied to NOR-gate 17 which provides the desired enable signal to AND gates 21.1 through 21.5 in bit-error detection and correction unit 20. These AND gates also receive the error vector pattern which for symbol position \( j \) as represented by syndrome subvector \( S_z \), \( \tilde{S}_z = (s_1 + s_2 + s_3 + s_4) \), or using the notation shown in FIG. 6A by \( S_z = (\tilde{S}_z, \tilde{S}_z, \tilde{S}_z, \tilde{S}_z) \). The error pattern is supplied to AND gates 21.1 through 21.5 respectively which are enabled by the output from multiple input NOR-gate 17. The outputs of AND gates 21.1 through 21.5 are, as described above, supplied to Exclusive-OR gates 22.1 through 22.5 which act as conditional inverters for received data symbol bits \( D_j \) through \( D_7 \). Thus, the output of bit error and correction unit 20 is the corrected symbol bits for symbol position \( j \). Similar decoding circuitry is provided for each of the other symbol positions. Furthermore, an uncorrectable error indication is provided through the operation of functional block 30 as shown in FIG. 6E.

Next is considered the specific relation between the parity check matrix shown in FIG. 7 and the circuit implementation thereof illustrated in FIG. 8B. More particularly, FIG. 7 represents a modification of the binary check matrix of FIG. 4. The matrix of FIG. 4 contains 90 binary columns. However, for a particular implementation only a 79 bit code was desired which included 64 information bits and 15 check bits. This particular organization resulted from an arrangement of memory chips organized in a 9 bit per chip configuration. In each memory access it was desired to retrieve 2x79 output bits from 18 memory chips the output being split into two error correctable words each with 64 information bits. Each ECC word is therefore divided into eleven 4 bit symbols and seven 5 bit symbols (79 bit total) as suggested by the bit count along the bottom of FIG. 7. This matrix, now even further reduced, was produced by deleting one binary column from eleven of the symbol columns. The rule chosen for the deletion was to remove the binary column having the largest weight. FIG. 7 shows the resultant parity check matrix for the error correction code that is therefore designed to correct errors generated from all single chip (symbol) failures and to detect errors generated from all double chip failures. The error correction circuits for the first symbol (\( j = 1 \)) are shown in FIG. 8B. In particular, it is seen that additional simplifications result, most notably with respect to syndrome bits \( s_8 \).

However, other simplifications result from the elimination of other inputs to Exclusive-OR gates 16.1 through 16.10. Moreover, fewer AND gates and Exclusive-OR gates are required in bit error detection and correction unit 20.

One of the advantages of the circuitry shown in FIGS. 8A and 8B is that the matrix-vector product and comparison operations can be carried out in a single layer of Exclusive-OR gates. It is recognized however that logic designers often employ other gates to implement Exclusive-OR functions. When implemented in terms of such gates particularly when complement and true forms of input variables are not available, it is recognized that two layers of logic may be required. Nonetheless this still represents a significant advantage in computation and circuit simplification.

It is further recognized that those skilled in the art of error correction code design that it is possible to derive other codes from a given parity check matrix. As indicated above, some of these codes are obtained by deleting columns of a parity check matrix. Accordingly, the present invention is also seen to be directed to coding systems which are derivable from the parity check matrix modifications of the present invention. In this regard, it is noted that in the process of deleting binary columns, particularly as seen in the transition from FIG. 4 to FIG. 7, matrices that are otherwise identity matrices are created but may have an extra row. Nonetheless, these matrices are meant to be included in applicants’ embodiments and those claims which indicate the presence of an identity sub-matrix in a symbol column.

Accordingly, it is seen that the present applicant has provided a method and apparatus for coding and decoding which results in low cost circuitry and improved decoding times. It is further seen that the present applicant has employed hitherto unknown techniques to manipulate and improve the performance of a large class of codes which are well known and which have highly useful error correcting properties. It is further seen that the applicant has implemented otherwise complex matrix-vector product and comparison operations in a single layer of circuitry, thus enhancing speed of operation while nonetheless providing code manipulation methods for insuring that the present codes can be implemented using minimal hardware.

While the invention has been described in detail herein in accordance with certain preferred embodiments thereof, many modifications and changes therein may be effected by those skilled in the art. Accordingly, it is intended by the appended claims to cover all such modifications and changes as fall within the true spirit and scope of the invention.

The invention claimed is:

1. A method for encoding input digital electrical signals into blocks of \( n \) bits and subblocks of at most \( b \) symbol bits, said method comprising the steps of:

   a. generating check bit signals from said input digital electrical signals so that said check bit signals satisfy parity check conditions determined by a binary parity check matrix derivable from a matrix of the form \( H = [H_1, H_2, \ldots, H_c] \) wherein each symbol column, \( H_j \), \( j = 1, 2, \ldots, N \), is a column of \( M \) disjoint binary submatrices having \( b \) rows and at most \( b \) columns and wherein at least
1. The method of claim 1 in which all of said N symbol columns possess a minimal number of ones.

2. The method of claim 1 in which M=3.

3. The method of claim 1 in which M=3.

4. The method of claim 1 in which each symbol column possesses at least one identity matrix.

5. The method of claim 1 in which each symbol column is either a matrix of zeros or a power of a matrix \( A \) which is the companion matrix associated with a primitive polynomial over \( GF(2) \).

6. The method of claim 5 in which the primitive polynomial is \( p(x)=x^4+x+1 \).

7. The method of claim 5 in which M=3 and in which, at least one symbol column, each symbol matrix is distinct and selected from the group consisting of the matrices \( I, T^2 \) and \( T^4 \) being an integer.

8. The method of claim 3 in which M=3 and in which, at each symbol column, each symbol matrix is distinct and selected from the group consisting of the matrices \( I, T \) and \( T^2 \) being an integer.

9. A method for decoding electrical signals represented by a linear binary code occurring in blocks of \( n \) bits and subblocks of at most \( b \) symbol bits, said electrical signals having been encoded subject to parity check conditions determined by a parity check matrix \( H \) of the form \( [H_1, H_2, \ldots, H_M] \), wherein each symbol column \( H_j \), \( j=1, 2, \ldots, M \), is a column of \( M \) disjoint binary submatrices \( H_{ij} \) and wherein, for at least one \( i \), \( H_{ij} \) includes an identity matrix with \( b \) rows and at most \( b \) columns, said decoding method comprising the steps of:

   - Determining a syndrome vector \( S \) for a received binary vector \( y \), said syndrome \( S \) having \( M \) subvectors \( S_1, S_2, \ldots, S_M \),

   - Determining, for at least one symbol position \( j \), the \( (M-1) \) matrix-vector products \( H_j S_i \), \( i \neq k \), where \( k \) denotes the matrix row of \( H_j \) which includes said identity matrix;

   - Comparing said \( (M-1) \) matrix-vector products \( H_j S_i \) with corresponding syndrome vectors \( S_i \) to determine the presence of correctable symbol errors; and

   - Correcting at least said \( j \)th symbol using the vector pattern \( S_j \) upon indication of correctable symbol errors from said comparison.

10. The decoding method of claim 9 which includes correcting all of said symbols.

11. The method of claim 9 in which there is a minimal number of ones amongst the binary elements in \( H_{ij}, H_{ik}, \ldots, H_{ij} \) for some \( j \) greater than 1 and less than or equal to \( M \).